

APPLICATION FOR UNITED STATES LETTERS PATENT

Title

SRC with Multiple Sets of Filter Coefficients in Memory and a High Order Coefficient Interpolator

Inventor(s):

Daniel L. W. Chieng, Jack B. Andersen, Larry E. Hand

Date Filed:

March 19, 2004

Attorney Docket No.:

D2A1230-1

Filed By:

Law Offices of Mark L. Berrier  
1250 Capital of Texas Highway South  
Building II, Suite 216  
Austin, Texas 78746  
Attn: Mark L. Berrier  
Tel. 512-329-2655  
Fax. 512-330-1710

USPS Express Mail Label No. :

EU826440602US

**SRC with Multiple sets of Filter Coefficients in Memory  
and a High Order Coefficient Interpolator**

**Related Applications**

This application claims priority to U.S. Provisional Patent Application No. 60/469,735, entitled "SRC with Multiple Sets of Filter Coefficients in Memory and a High Order Coefficient Interpolator," by Chieng, et al., filed May 12, 2003; U.S. Provisional Patent Application No. 60/456,414, entitled "Adaptive Anti-Clipping Protection," by Taylor, et al., filed March 21, 2003; U.S. Provisional Patent Application No. 60/456,430, entitled "Frequency Response Correction," by Taylor, et al., filed March 21, 2003; U.S. Provisional Patent Application No. 60/456,429, entitled "High-Efficiency, High-Performance Sample Rate Converter," by Andersen, et al., filed March 21, 2003; U.S. Provisional Patent Application No. 60/456,421, entitled "Output Device Switch Timing Correction," by Taylor, et al., filed March 21, 2003; U.S. Provisional Patent Application No. 60/456,422, entitled "Output Filter, Phase/Timing Correction," by Taylor, et al., filed March 21, 2003; U.S. Provisional Patent Application No. 60/456,428, entitled "Output Filter Speaker/Load Compensation," by Taylor, et al., filed March 21, 2003; U.S. Provisional Patent Application No. 60/456,420, entitled "Output Stage Channel Timing Calibration," by Taylor, et al., filed March 21, 2003; U.S. Provisional Patent Application No. 60/456,427, entitled "Intelligent Over-Current, Over-Load Protection," by Hand, et al., filed March 21, 2003; each of which is fully incorporated by reference as if set forth herein in its entirety.

## **Background of the Invention**

[0001] Field of the invention.

[0002] The invention relates generally to audio amplification systems, and more particularly to systems and methods for converting data streams at a first sample rate to a second sample rate using a coefficient interpolator that interpolates selectable sets of coefficients.

[0003] Related art.

[0004] Pulse Width Modulation (PWM) or Class D signal amplification technology has existed for a number of years. PWM technology has become more popular with the proliferation of Switched Mode Power Supplies (SMPS). Since this technology emerged, there has been an increased interest in applying PWM techniques in signal amplification applications as a result of the significant efficiency improvement that can be realized through the use of Class D power output topology instead of the legacy (linear Class AB) power output topology.

[0005] Early attempts to develop signal amplification applications utilized the same approach to amplification that was being used in the early SMPS. More particularly, these attempts utilized analog modulation schemes that resulted in very low performance applications. These applications were very complex and costly to implement. Consequently, these solutions were not widely accepted. Prior art analog implementations of Class D technology have therefore been unable to displace legacy Class AB amplifiers in mainstream amplifier applications.

- [0006] Recently, digital PWM modulation schemes have surfaced. These schemes use Sigma-Delta modulation techniques to generate the PWM signals used in the newer digital Class D implementations. These digital PWM schemes, however, did little to offset the major barriers to integration of PWM modulators into the total amplifier solution. Class D technology has therefore continued to be unable to displace legacy Class AB amplifiers in mainstream applications.
- [0007] One of the problems with conventional implementations of Class D technology lies in the conversion of the digital input data from an input sample rate to an internal sample rate. This process usually involves upsampling and then downsampling the input audio signal to achieve the desired sample rate and filtering the signal either during the up/downsampling or thereafter.
- [0008] The upsampling and downsampling of the input audio signal may be performed using a polyphase filter. Rather than generating a large number of samples and then throwing away unneeded samples, the polyphase filter generates only those samples that will be retained. In order to achieve good performance, however, it is typically necessary to store  $2^{18}$  filter coefficients that define the filter. It is impractical to store this large number of coefficients. Further, when designing such a filter for this purpose, there are inherent tradeoffs between, for example, passband ripple, stopband rejection and rolloff. Because of the tradeoffs, it is difficult to design a filter that is optimal to cover multiple applications. It would likewise be impractical to have change all of these coefficients to modify the frequency response characteristics for the filter.
- [0009] A sample rate converter (SRC) based on multirate signal processing requires an interpolation filter. When designing such an FIR low pass filter, there are inherent tradeoffs between passband ripple, stopband rejection and rolloff. This makes it hard to determine an optimal filter to cover multiple applications.

## Summary of the Invention

- [0010] One or more of the problems outlined above may be solved by the various embodiments of the invention. Broadly speaking, the invention comprises systems and methods for converting a data stream from a first sample rate to a second sample rate using a sample rate converter that employs selectable filters. The filters are implemented by providing multiple sets of filter coefficients in a memory, selecting one of the sets of filter coefficients and performing coefficient interpolation to produce filter coefficients that are convolved with the input data stream to produce a re-sampled output data stream.
- [0011] One embodiment comprises a method including storing a plurality of sets of filter coefficients in a memory, selecting a first one of the sets of filter coefficients, interpolating the first selected set of filter coefficients and convolving the interpolated first selected filter coefficients with an input signal to produce a filtered output signal. In one embodiment, the input signal is an audio signal that is convolved with interpolated polyphase filter coefficients in the sample rate converter of a digital audio amplifier such as a PWM amplifier. In one embodiment, the set of filter coefficients is selected according to a value that is stored in a filter selection register. The value in the register may be set by a DSP or by a user. In one embodiment, the sets of filter coefficients are all stored in a single memory and are interpolated according to a cubic spline interpolation algorithm.
- [0012] An alternative embodiment of the invention comprises a system including a coefficient interpolator and a memory coupled to the coefficient interpolator,

where the memory is configured to store multiple sets of filter coefficients and the coefficient interpolator is configured to interpolate a selected one of the sets of filter coefficients. In one embodiment, the system includes a filter selection register that stores a filter selection value. In this embodiment, the coefficient interpolator is configured to select the set of filter coefficients indicated by this filter selection value. The filter selection value in the register may be modified by a user or by a DSP within the system. In one embodiment, the system may include a convolution engine configured to convolve an input signal with the interpolated coefficients to produce an output signal. In one embodiment, the system is implemented in a sample rate converter of a PWM amplifier. In one embodiment, the sets of filter coefficients are all stored in a single memory and are interpolated according to a cubic spline interpolation algorithm.

[0013] Numerous additional embodiments are also possible.

### **Brief Description of the Drawings**

- [0014] Other objects and advantages of the invention may become apparent upon reading the following detailed description and upon reference to the accompanying drawings.
- [0015] FIGURE 1 is a functional block diagram illustrating a digital audio amplification system using PWM technology.
- [0016] FIGURE 2 is a diagram illustrating the manner in which sample rate conversion is typically performed.
- [0017] FIGURE 3 is a diagram illustrating the interpolation and decimation of a sampled input signal to produce a corresponding signal at a different sample rate.
- [0018] FIGURE 4 is a diagram illustrating the components of a sample rate converter in accordance with one embodiment of the invention.
- [0019] FIGURE 5 is a diagram illustrating the storage of multiple, different sets of filter coefficients in a memory for use by an interpolator in a sample rate converter in accordance with one embodiment.
- [0020] While the invention is subject to various modifications and alternative forms, specific embodiments thereof are shown by way of example in the drawings and the accompanying detailed description. It should be understood, however, that the drawings and detailed description are not intended to limit the invention to the particular embodiment which is described. This disclosure is instead

intended to cover all modifications, equivalents and alternatives falling within the scope of the present invention as defined by the appended claims.

## Detailed Description of Preferred Embodiments

- [0021] One or more embodiments of the invention are described below. It should be noted that these and any other embodiments described below are exemplary and are intended to be illustrative of the invention rather than limiting.
- [0022] As described herein, various embodiments of the invention comprise systems and methods for converting a data stream from a first sample rate to a second sample rate using a sample rate converter that employs selectable filters. The filters are implemented by providing multiple sets of filter coefficients in a memory, selecting one of the sets of filter coefficients and performing coefficient interpolation to produce filter coefficients that are convolved with the input data stream to produce a re-sampled output data stream.
- [0023] A sample rate converter (SRC) based on multirate signal processing requires an interpolation filter. When designing such an FIR low pass filter, there are inherent tradeoffs between passband ripple, stopband rejection and rolloff. This makes it hard to determine an optimal filter to cover multiple applications. One solution to this problem is to keep multiple filters in the memory (ROM or RAM). Each set of filter coefficients is selectable by the software via a register setting, allowing the digital signal processor (DSP) to choose the best filter for a particular application. To achieve good performance, filter coefficients in the order of  $2^{18}$  have to be stored for each filter. This is impractical for a single filter, much less multiple filters. A solution to memory size reduction is to store a subset of coefficients and use high order interpolator to interpolate actual coefficients.

- [0024] Existing SRCs may store one set of FIR filter coefficients, or one set of filter coefficients plus the difference of two adjacent coefficients (to aid in linear interpolation), or have filters for a multi-stage SRC.
- [0025] In one embodiment of the present invention, four sets of filter coefficients are stored in a single ROM. A high order interpolator is based on a cubic spline approximation. The exceptional accuracy of this approximation makes it a good choice despite its relatively complex implementation. This invention may be extended to storing any number of sets of filter coefficients in the coefficient memory.
- [0026] A preferred embodiment of the invention is implemented in an audio amplification system. As noted above, pulse width modulation (PWM) technology has recently been applied in audio amplification systems, but has suffered from the drawbacks of conventional methodologies. These methodologies employ analog modulation schemes which are very complex and costly, and which provide relatively poor performance. The present systems and methods are instead implemented in digital modulation schemes and employ methodologies which overcome some of the problems that existed in the prior art.
- [0027] Referring to FIGURE 1, a functional block diagram illustrating a digital audio amplification system using PWM technology is shown. In this embodiment, system 100 receives a digital input data stream from a data source such as a CD player, MP3 player, digital audio tape, or the like. The input data stream is received by sample rate converter 110. The input data stream has a particular sample rate which depends upon the data source. This sample rate is typically one of a set of predetermined sample rates that are used by the corresponding type of device. For example, a CD player may output digital data with a sample

rate of 44.1 kHz, while a digital audio tape player may output data with a sample rate of 32 kHz.

[0028] In the present systems and methods, sample rate converter 110 converts the input data stream from the sample rate at which it was received to a predetermined internal rate which is used within system 100. In one embodiment, this internal sample rate is 100 kHz. Thus, if data is received at a sample rate of 50 kHz, sample rate converter 110 will re-sample the data to produce a corresponding internal data stream at a sample rate of 100 kHz. This internal data stream is then provided to an audio effects subsystem 120. Audio effects subsystem 120 performs any desired processing on the internal data stream and provides the resulting processed data stream to PWM modulator 130.

[0029] The data stream received by PWM modulator 130 represents a pulse code modulated signal. PWM modulator 130 converts this data stream to a pulse width modulated signal. The pulse width modulated signal is then provided to output stage 140. In output stage 140 amplifies the pulse width modulated signal and may perform some filtering or further processing of the amplified signal. The resulting signal is then output to a speaker system 150, which converts the electrical signal to an audible signal which can be heard by a listener.

[0030] The present disclosure focuses on the sample rate converter in the audio system described above. As explained above, the purpose of the sample rate converter is to receive an input data stream which is sampled at a first rate, and to generate an output data stream which is sampled at a second rate. While the audio signal which is represented by the data stream remains essentially unchanged (at least in some embodiments), the sampling rate is changed to

conform to the requirements of the audio system so that it can be processed by the system.

[0031] Referring to FIGURE 2, a diagram illustrating the manner in which sample rate conversion is typically performed is shown. As depicted by this figure, an input data stream is first up-sampled, or interpolated, by a first filter 210, and is down-sampled, or decimated, by a second filter 220. An intermediate filter 230 is used to low-pass filter the up-sampled data before it is decimated. The input data stream has a first sample rate,  $F_{in}$ . This data stream is up-sampled by a factor of  $M$ . Thus, after up-sampling, the data stream has a sample rate of  $M \times F_{in}$ . The up-sampling is typically achieved by interpolating between the samples of the input data stream to generate intermediate samples.  $M$  is chosen so that the intermediate sample rate ( $M \times F_{in}$ ) is higher than the desired output sample rate,  $F_{out}$ . Typically, the intermediate rate is much higher than the desired output rate.

[0032] The up-sampled data stream is low-pass filtered and then decimated to reduce the sample rate from the intermediate rate to the desired output rate. After down-sampling, the sample rate is  $F_{out} = (M/N) \times F_{in}$ . The down-sampling, or decimation, of the data stream is typically accomplished by dropping samples from the intermediate data stream. For example, if the intermediate data stream is sampled at 200 kHz and the desired output sample rate is 100 kHz, every other sample will be dropped.

[0033] Ideally,  $M$  and  $N$  are integers. If  $M$  is an integer, the up-sampling of the input data stream comprises inserting  $M-1$  new samples, evenly spaced between each of the original samples. Then, if  $N$  is an integer, the down-sampling of the intermediate data stream comprises taking only every  $N$ th sample and dropping the rest. This is illustrated in FIGURE 3.

[0034] FIGURE 3 is a diagram illustrating the interpolation and decimation of a sampled input signal to produce a corresponding signal at a different sample rate. In this figure, the input samples are represented by points 301, 306, 311 and 316. The straight-line interpolated value of the signal is represented by the dotted lines. The signal is up-sampled by a factor of 5, so 4 additional sample points are interpolated between each pair of adjacent samples. Thus, points 302-305 are inserted in the interval between sample 301 and sample 306. Likewise, points 307-310 are inserted between samples 306 and 311, and points 312-315 are inserted between samples 311 and 316. After being low-pass filtered, the resulting points (301-316) are down-sampled by a factor of 3, so every third point is used, and the remainder are discarded. The resulting data stream consists of samples 301, 304, 307, 310, 313 and 316 (as indicated by the arrows).

[0035] One of the problems with a straightforward implementation of the up-sampling and down-sampling of the input data stream is that, in order to make  $M$  and  $N$  integers, and in order to maintain the desired resolution,  $M$  and  $N$  typically must be very large numbers. Consider the example of FIGURE 3. If  $F_{in}$  is 60 kHz and  $F_{out}$  is 100 kHz,  $M$  is 5 and  $N$  is 3. If  $F_{in}$  were 60.5 kHz instead of 60 kHz, however it would be necessary to select  $M = 200$  and  $N = 121$ . Scenarios requiring even higher values for  $M$  and  $N$  can easily be developed. Based upon the resolution of the sample rate converter in the preferred embodiment, values of up to  $2^{18}$  might be necessary.

[0036] Another problem with the interpolation-and-decimation methodology is that it may be difficult to handle variations in the sample rates of the received data streams. In typical audio systems, each device or component may generate its own clock signal upon which the corresponding sample rate is based. Even if the clock signals for two components are intended to be identical, however, the clock signals are not synchronized and may have slight variations. As a result

of the differences in clock signals, data may be dropped, or buffers may overflow, resulting in errors. The present sample rate converter is designed to handle these differences.

[0037] It should be noted that audio systems may also include various different types of audio sources. For example, the audio signal may be generated by a CD player, MP3 player, digital audio tape or the like. These devices may be configured to generate audio signals at different sample rates. For instance, a CD player may provide an output signal that has a 44.1 kHz sample rate, while a digital audio tape player may generate an output signal at a 32 kHz sample rate. The present systems and methods enable the sample rate converter to accommodate multiple different sample rates in the input data stream. Moreover, the sample rate converter is capable of independently adjusting each channel to accommodate a different input sample rate. By comparison, prior art systems can only accommodate different sample rates on different channels if the two sample rates are known.

[0038] The accommodation of different sample rates, and variations between rates that are nominally the same, may be achieved through the use of a polyphase filter. The polyphase filter performs the functions of both interpolator 210 and decimator 220. The polyphase filter performs these functions by interpolating the input data stream in a manner which does not require that the data stream be up-sampled by an integer factor or down-sampled by an integer factor.

[0039] The interpolator and the decimator described above are typically implemented as (FIR-type) filters. The polyphase filter is obviously also a filter, but rather than generating a large number of samples (as performed by the interpolation filter) and then throwing away unneeded samples and (as performed by the decimation filter), the polyphase filter generates only those samples that will, in the end, be retained. Thus, compared to the example of FIGURE 3, rather than

generating samples 301-316 and then discarding two-thirds of these samples, only samples 301, 304, 307, 310, 313 and 316 are generated, and none are discarded.

[0040] The polyphase filter is defined by a set of filter coefficients. If the coefficients are extrapolated to a different set of coefficients, different sampling rates are achieved. This enables non-integer sample rate conversion through the choice of appropriate filter coefficients.

[0041] A typical sample rate converter that uses polyphase filters contains memory for a single set of filter coefficients. Because the present systems and methods use an interpolator to generate appropriate coefficients for convolution with the samples of the input data stream, a greatly reduced number of coefficients can be stored in memory for use by the interpolator. This, in turn, reduces the amount of memory required for storage of the coefficients. It is therefore practical to store coefficients corresponding to several different filters in the coefficient memory.

[0042] In one embodiment, four sets of coefficients corresponding to four different filters are stored in the memory. Referring to FIGURE 5, a diagram illustrating the use of the different filter coefficients (hence different filters) is shown. In this embodiment, an address generator 510 is used to select the appropriate set of filter coefficients from memory 530 by controlling a multiplexer 520 that provides the coefficients to the interpolator 540. In one embodiment, address generator 510 also controls selection of particular coefficients within each set for use by interpolator 540. Interpolator 540 and coefficient memory 530 are integrated into the sample rate converter as described in relation to FIGURE 4 (where the interpolator and memory are combined in the figure as interpolation unit 460).

[0043] In one embodiment, the selection of one of the sets of filter coefficients is based upon the contents of a filter selection register. In the case of a system that has

four available sets of filter coefficients, the filter selection register may contain a simple, 2-bit value that can range from 0-3. In the embodiment described above, address generator 510 may examine the contents of the filter selection register as part of the process of generating addresses for controlling multiplexer 520. The contents of the filter selection register may be automatically updated by a DSP internal to the system, or they may be altered manually through user input to the system.

[0044] Referring to FIGURE 4, a diagram illustrating the components of a sample rate converter in accordance with one embodiment of the invention is shown. The lower half of the figure generally corresponds to a data path for the audio data that will be converted, while the upper half of the figure corresponds to a control path for controlling the actual sample rate conversion. The control path includes the interpolator and the filter coefficient memory.

[0045] As shown in FIGURE 4, samples of an audio data stream are received and stored in an input FIFO 405. The input data stream has a sample rate of  $F_{in}$ . The samples are read from FIFO 405 and convolved with a set of interpolated coefficients by convolution engine 410. Convolution engine 410 effectively up-samples or down-samples the data to produce samples at a rate equivalent to the output rate ( $F_{out}$ ) of the sample rate converter. These samples are stored in an output FIFO 406. The samples are then read out of output FIFO 406 at rate  $F_{out}$ .

[0046] Frame sync signals associated with the audio data are received by rate estimator counters 421 and 422. Rate estimator counters 421 and 422 simply count the numbers of clock cycles between samples received on the respective channels. (It should be noted that, while the present embodiment has two channels, and corresponding rate estimators, other embodiments may handle N channels and have N corresponding sets of components.) One of the rate

estimator counters is selected by multiplexer 430 and the corresponding count is filtered by low pass filter 440. The filtered sample rate count is forwarded to phase selection unit 450, and is used to interpolate the filter coefficients for the polyphase filter. The interpolated polyphase filter coefficients are then convolved with the data samples in convolution unit 410 to produce the re-sampled data.

[0047] The flow of data samples through FIFO 405 and FIFO 406 are managed by FIFO management unit 407. based on the flow of data, FIFO management unit 407 provides feedback to feedback unit 470. This feedback is used to adjust low pass filter 440. Effectively, this adjusts the sample rate which is estimated and thereby adjusts the coefficient interpolation performed in the sample rate converter. The sample rate conversion is thereby also adjusted to more closely track the actual input sample rate and to prevent the overflow or underflow of FIFOs 405 and 406.

[0048] In one embodiment, rate estimator counters 421 and 422 are 24-bit counters. Each can select from four input frame sync signals: SAI LRCK; SPDIF RX frame sync; Packet Data frame sync; and ESSI frame sync. The period measurement is accomplished by counting the number of DSP clock cycles within the counting period of the frame sync signal. The counting period is programmable, typically with the period equal to 1. In this embodiment, the count is multiplied by a gain. The gain is a 12-bit integer which is typically set to a power of 2, which is equivalent to a 1-bit left shift which creates an additional bit of resolution.

[0049] Low pass filter 440 is, in one embodiment, a second-order IIR filter. This filter may, for example, comprise a pair of cascaded first-order IIR filters. Low pass filter 440 attenuates jitter in the count received from the rate estimator counter. This ensures that the count changes slowly, and thereby improves the quality of

the sample rate conversion. The averaging process that is implemented by the low pass filter causes the potential for buffer underflow or overflow. This problem is corrected by implementing closed loop feedback in the software which adjusts a 24-bit offset that is added to the count value before the value is passed through low pass filter 440. In one embodiment, the filter coefficient of low pass filter 440 is adjustable to allow faster frequency and phase lock.

[0050] Coefficient interpolator 460 works in conjunction with the ROM in which the coefficients are stored and the ROM address generator that provides addresses for retrieval of the coefficients for use by the interpolator. The filter coefficients are actually stored in two ROMs -- one stores even coefficients, while the other stores odd coefficients. Each of these two ROMs has corresponding halves of the multiple, selectable sets of filter coefficients. The interpolator performs a cubic spline interpolation. The interpolator employs a five-stage, two-cycle pipeline to perform the interpolation, thereby enabling resource sharing while maintaining a throughput of one interpolation per two clock cycles.

[0051] In one embodiment, the software of the sample rate converter is responsible for performing a number of tasks. For example, as mentioned above, rate estimator counters 421 and 422 multiply their respective counter values by a gain, but the gain is determined by the software. Similarly, the offset and filter coefficients for the low pass filter following the rate estimator counters are determined by the software. The software is further responsible for calculating the ratio of the input sample rate ( $F_{in}$ ) to the output sample rate ( $F_{out}$ ). Based upon the ratio of sample rates and the filtered counter values, the software determines the filter length, phase and phase increment for interpolation of the polyphase filter coefficients. Further, the software is responsible for convolving the polyphase filter coefficients with the input samples, managing the input and output FIFOs, and providing feedback for adjustment of the estimated input sample rate.

- [0052] The software components are implemented in a data processor. Typical modern processors have the capability of executing tight loops very efficiently while reading in data streams. For example, digital signal processors (DSP's) have "zero overhead looping" capability. Modern microcontrollers also have the capability of executing multiple instructions per cycle. These DSP's and microcontrollers typically also have separate program and data memories that make them suitable for sample rate converter applications.
- [0053] These processors have the capability, for example, to execute the following in one processor cycle: read a data sample from memory (as indicated by a sample pointer register); update the sample pointer register to point to a next sample; multiply the data sample by a coefficient value; and add (accumulate) the result of the multiplication in a data register. If the polyphase filter contains  $X$  coefficients,  $X$  clock cycles are used to compute one output sample.
- [0054] Aside from the functions mentioned above, the sample rate converter is responsible for reading and interpolating the polyphase filter coefficients. A processor can handle a number of parallel channels  $Y$  at the same time, where  $Y$  is limited by the available number of accumulator and sample pointer registers. When  $Y$  channels are processed simultaneously using identical coefficients, relatively compact hardware can be designed to perform the following in  $Y$  or less cycles: read a number of coefficients from memory (as indicated by coefficient pointer); update a coefficient pointer register; and perform interpolation to calculate filter coefficients to a desired precision.
- [0055] In "pseudo C" the processor would do the following:

for every output sample

Initialize the hardware coefficient calculator

```
for j=1 to Y
    o[Y] = 0;                // Initialize accumulators
    p[Y] = start(N);        // Initialize pointers

for i=1 to X                // For every coefficient
    C = mem[coeff]          // Read coefficient
    for j=1 to Y            // For every channel
        o[Y] += C*mem[p[Y]++]
```

[0056] Typically, the inner loop using j would be unrolled, and reading the next coefficient would be done in parallel with the last iteration (j=Y). A simple and efficient processor would calculate a new coefficient for every Y cycles. A more flexible solution would calculate a coefficient in Y or fewer cycles. When a new sample becomes available, it will halt computations until this sample is read and thereby automatically adjust to the rate at which the DSP reads the filter coefficients. Besides making the actual value of Y more flexible, this also allows the processor to periodically halt the computations and service other functions like interrupts.

[0057] In some embodiments, components of the sample rate converter may be shared between two or more independent sample rate conversion paths. For instance, two different paths may both use the same interpolator hardware. Each of these paths may use the same set of polyphase filter coefficients, or different sets of coefficients, depending upon the implementation.

[0058] Those of skill in the art will understand that information and signals may be represented using any of a variety of different technologies and techniques. For example, data, instructions, commands, information, signals, bits, symbols, and chips that may be referenced throughout the above description may be

represented by voltages, currents, electromagnetic waves, magnetic fields or particles, optical fields or particles, or any combination thereof. The information and signals may be communicated between components of the disclosed systems using any suitable transport media, including wires, metallic traces, vias, optical fibers, and the like.

[0059] Those of skill will further appreciate that the various illustrative logical blocks, modules, circuits, and algorithm steps described in connection with the embodiments disclosed herein may be implemented as electronic hardware, computer software, or combinations of both. To clearly illustrate this interchangeability of hardware and software, various illustrative components, blocks, modules, circuits, and steps have been described above generally in terms of their functionality. Whether such functionality is implemented as hardware or software depends upon the particular application and design constraints imposed on the overall system. Those of skill in the art may implement the described functionality in varying ways for each particular application, but such implementation decisions should not be interpreted as causing a departure from the scope of the present invention.

[0060] The various illustrative logical blocks, modules, and circuits described in connection with the embodiments disclosed herein may be implemented or performed with general purpose processors, digital signal processors (DSPs) or other logic devices, application specific integrated circuits (ASICs), field programmable gate arrays (FPGAs), discrete gates or transistor logic, discrete hardware components, or any combination thereof designed to perform the functions described herein. A general purpose processor may be any conventional processor, controller, microcontroller, state machine or the like. A processor may also be implemented as a combination of computing devices, e.g., a combination of a DSP and a microprocessor, a plurality of

microprocessors, one or more microprocessors in conjunction with a DSP core, or any other such configuration.

[0061] The steps of the methods or algorithms described in connection with the embodiments disclosed herein may be embodied directly in hardware, in software or firmware modules executed by a processor, or in a combination thereof. A software product may reside in RAM memory, flash memory, ROM memory, EPROM memory, EEPROM memory, registers, hard disk, a removable disk, a CD-ROM, or any other form of storage medium known in the art. An exemplary storage medium is coupled to the processor such the processor can read information from, and write information to, the storage medium. In the alternative, the storage medium may be integral to the processor. The processor and the storage medium may reside in an ASIC. The ASIC may reside in a user terminal. In the alternative, the processor and the storage medium may reside as discrete components in a user terminal.

[0062] The previous description of the disclosed embodiments is provided to enable any person skilled in the art to make or use the present invention. Various modifications to these embodiments will be readily apparent to those skilled in the art, and the generic principles defined herein may be applied to other embodiments without departing from the spirit or scope of the invention. Thus, the present invention is not intended to be limited to the embodiments shown herein but is to be accorded the widest scope consistent with the principles and novel features disclosed herein.

[0063] The benefits and advantages which may be provided by the present invention have been described above with regard to specific embodiments. These benefits and advantages, and any elements or limitations that may cause them to occur or to become more pronounced are not to be construed as critical, required, or essential features of any or all of the claims. As used herein, the

terms “comprises,” “comprising,” or any other variations thereof, are intended to be interpreted as non-exclusively including the elements or limitations which follow those terms. Accordingly, a system, method, or other embodiment that comprises a set of elements is not limited to only those elements, and may include other elements not expressly listed or inherent to the claimed embodiment.

[0064] While the present invention has been described with reference to particular embodiments, it should be understood that the embodiments are illustrative and that the scope of the invention is not limited to these embodiments. Many variations, modifications, additions and improvements to the embodiments described above are possible. It is contemplated that these variations, modifications, additions and improvements fall within the scope of the invention as detailed within the following claims.